AMENDMENT TO THE CLAIMS

All claims currently pending and under consideration in the above-identified application are shown below. Claim 21 is currently amended. This listing of claims will replace all prior versions and listings of claims in the above-identified application.

1. (Previously Presented) A computer implemented method for rendering a desktop window in a graphical user interface of an operating system shell, comprising:

receiving, at a composting desktop window manager (CDWM), application content from advanced applications in a bottom-to-top order, to display the application content received in a bottom-to-top order in windows corresponding to the advanced applications in the graphical user interface;

receiving, at a desktop window manager (DWM), application content from legacy applications in a top-to-bottom order to display the application content received in a top-to-bottom order in windows corresponding to the legacy application in the graphical user interface;

stripping out application content received from the legacy applications;

converting the stripped application content to a graphical representation;

switching between the CDWM and the DWM to render the advanced application content and legacy application content; and

displaying at least a portion of the application content in an opaque content portion of the windows, the windows having translucent frame portions.

2. (Previously Presented) The computer implemented method of claim 1, wherein the displaying step comprises a pixel shader content on top of which the frame portion is rendered.

3. (Withdrawn) The computer implemented method of claim 1, wherein the translucent frame portion of the window comprises a likeness of glass.

4. (Withdrawn) The computer implemented method of claim 1, wherein the translucent frame portion of the window comprises a likeness of frosted glass.

5-6. (Canceled).

7. (Previously Presented) The computer-implemented method of claim 1, wherein displaying comprises:

the CDWM modeling the window by applying a texture to a mesh.

8. (Original) The computer-implemented method of claim 7, wherein the mesh is defined by a current visual style.

9. (Original) The computer-implemented method of claim 7, wherein the mesh is provided in the application content information.

10. (Original) The computer-implemented method of claim 7, wherein the texture is defined by a current visual style.

11. (Original) The computer-implemented method of claim 7, wherein the texture is provided in the application content information.

12-14. (Canceled).

15. (Original) The computer-implemented method of claim 13, wherein the switching is based on the current visual style.

16. (Previously Presented) The computer-implemented method of claim 13, wherein the switching is based on a current power configuration of a computer on which the method is being performed.

17. (Original) The computer-implemented method of claim 1, wherein the frame comprises spectral highlights based on a virtual light source.

18. (Original) The computer-implemented method of claim 1, wherein the frame comprises reflective content based on other content in the graphical user interface separate from the window.

19. (Original) The computer implemented method of claim 1, wherein the frame portion is translucent when the window has an input focus.

20. (Original) The computer implemented method of claim 7 further comprising:

receiving user input to resize the window;

dividing the mesh into three regions per mesh dimension;

for each region, maintaining offsets of mesh vertices in any dimension by which the region is bounded by a bounding box of the window, and scaling mesh vertices in any dimension by which the region is not bounded by the bounding box of the window.

21. (Currently Amended) A computer storage medium storing computer executable instructions that cause a computer to perform a method for rendering a desktop window in a graphical user interface of an operating system shell, comprising:

receiving, at a [[composting]] compositing desktop window manager (CDWM), application content from advanced applications in a bottom-to-top order, to display the application content received in a bottom-to-top order in windows corresponding to the advanced applications in the graphical user interface;

receiving, at a desktop window manager (DWM), application content from legacy applications in a top-to-bottom order to display the application content received in a top-to-bottom order in windows corresponding to the legacy application in the graphical user interface, wherein the DWM redirects the application content received to the CDWM;

stripping out application content received from the legacy applications;

converting the stripped application content to a graphical representation; and

displaying at least a portion of the application content in an opaque content portion of the windows, the windows having translucent frame portions.

22. (Previously Presented) The computer storage medium of claim 21, wherein the displaying step comprises a pixel shader content on top of which the frame portion is rendered.

23. (Withdrawn-Previously Presented) The computer storage medium of claim 21, wherein the translucent frame portion of the window comprises a likeness of glass.

24. (Withdrawn-Previously Presented) The computer storage medium of claim 21, wherein the translucent frame portion of the window comprises a likeness of frosted glass.

25-26. (Canceled).

27. (Previously Presented) The computer storage medium of claim 21, wherein displaying comprises:

the CDWM modeling the window by applying a texture to a mesh.

28. (Previously Presented) The computer storage medium of claim 27, wherein the mesh is defined by a current visual style.

29. (Previously Presented) The computer storage medium of claim 27, wherein the mesh is provided in the application content information.

30. (Previously Presented) The computer storage medium of claim 27, wherein the texture is defined by a current visual style.

31. (Previously Presented) The computer storage medium of claim 27, wherein the texture is provided in the application content information.

32-36. (Canceled).

37. (Previously Presented) The computer storage medium of claim 21, wherein the frame comprises spectral highlights based on a virtual light source.

38. (Previously Presented) The computer storage medium of claim 21, wherein the frame comprises reflective content based on other content in the graphical user interface separate from the window.

39. (Previously Presented) The computer storage medium of claim 21, wherein the frame portion is translucent when the window has an input focus.

40. (Previously Presented) The computer storage medium of claim 27 further comprising:

receiving user input to resize the window;

dividing the mesh into three regions per mesh dimension;

for each region, maintaining offsets of mesh vertices in any dimension by which the region is bounded by a bounding box of the window, and scaling mesh vertices in any dimension by which the region is not bounded by the bounding box of the window.

41. (Previously Presented) A computer implemented method for rendering a desktop window in a graphical user interface of an operating system shell, comprising:

receiving application content to display in a window; and

displaying at least a portion of the application content in a content portion of the window having a frame portion, wherein the displaying further comprises

rendering spectral highlights on the frame portion based on a virtual light source by a compositing desktop window manager configured to provide transparency, shadows, lighting effects, bump mapping, and environmental mapping.

42. (Previously Presented) A computer implemented method for rendering a desktop window in a graphical user interface of an operating system shell, comprising:

receiving, at a compositing desktop window manager, application content in reverse z-order to display in a window; and

displaying at least a portion of the application content in a content portion of the window having a frame portion, wherein the displaying further comprises rendering reflective content on the frame portion based on other discrete content separate from the window in the graphical user interface by the compositing desktop window manager, wherein the compositing desktop window manager is configured to provide transparency, shadows, lighting effects, bump mapping, and environmental mapping.

43. (Previously Presented) A computer implemented method for rendering a desktop window in a graphical user interface of an operating system shell, comprising:

receiving, at a compositing desktop window manager, application content in reverse z-order to display in a window; and

displaying at least a portion of the application content in a content portion of the window having a frame portion, wherein the displaying further comprises

rendering refractive content on the frame portion based on other discrete content behind the window in the graphical user interface by the compositing desktop window manager, wherein the compositing desktop window manager is configured to provide transparency, shadows, lighting effects, bump mapping, and environmental mapping.

44. (Previously Presented) In a computer operating system that uses a composited desktop rendering model, a computer-implemented method of providing legacy support for applications compatible only with an invalidation desktop rendering model, the computer-implemented method comprising:

providing legacy window information from an instance of a legacy application program to a legacy desktop window manager (DWM) executing on the computer;

stripping out client content from the legacy window information;

converting the client content to a raster image of the client content,

a compositing desktop window manager (CDWM), executing on the computer, drawing a window to a buffer memory, wherein the CDWM renders the window by applying a texture to a mesh, and wherein the texture comprises the raster image of the client content and default non-client information.

45. (Previously Presented) A computer-implemented method for resizing a window defined in part by a mesh, the computer-implemented method comprising:

dividing the mesh, associated with the window displayed by the computer, into three regions per mesh dimension;

for each region, maintaining offsets of mesh vertices in any dimension by which the region is bounded by a bounding box of the window, and scaling mesh vertices in any dimension by which the region is not bounded by the bounding box of the window.

46. (Previously Presented) The computer-implemented method of claim 45, wherein the regions are equally sized.

47. (Previously Presented) The computer-implemented method of claim 45, wherein the regions are not equally sized.

48. (Previously Presented) The computer-implemented method of claim 45, wherein regions bounded by the bounding box are as small as necessary to encompass material that should not be scaled.